

Angular vs Aurelia



Steve Hartzog

Chief Scientist



<http://about.me/SteveHartzog>

- ▶ Dark matter developer
 - ▶ ~25 years of systems & dev experience
 - ▶ TypeScript, ECMAScript
 - ▶ Gulp, Grunt, Webpack
 - ▶ Node, AngularJS, Aurelia
 - ▶ Classic ASP, ASP.NET, C#, VB
 - ▶ SQL, Oracle, IIS, Apache

“

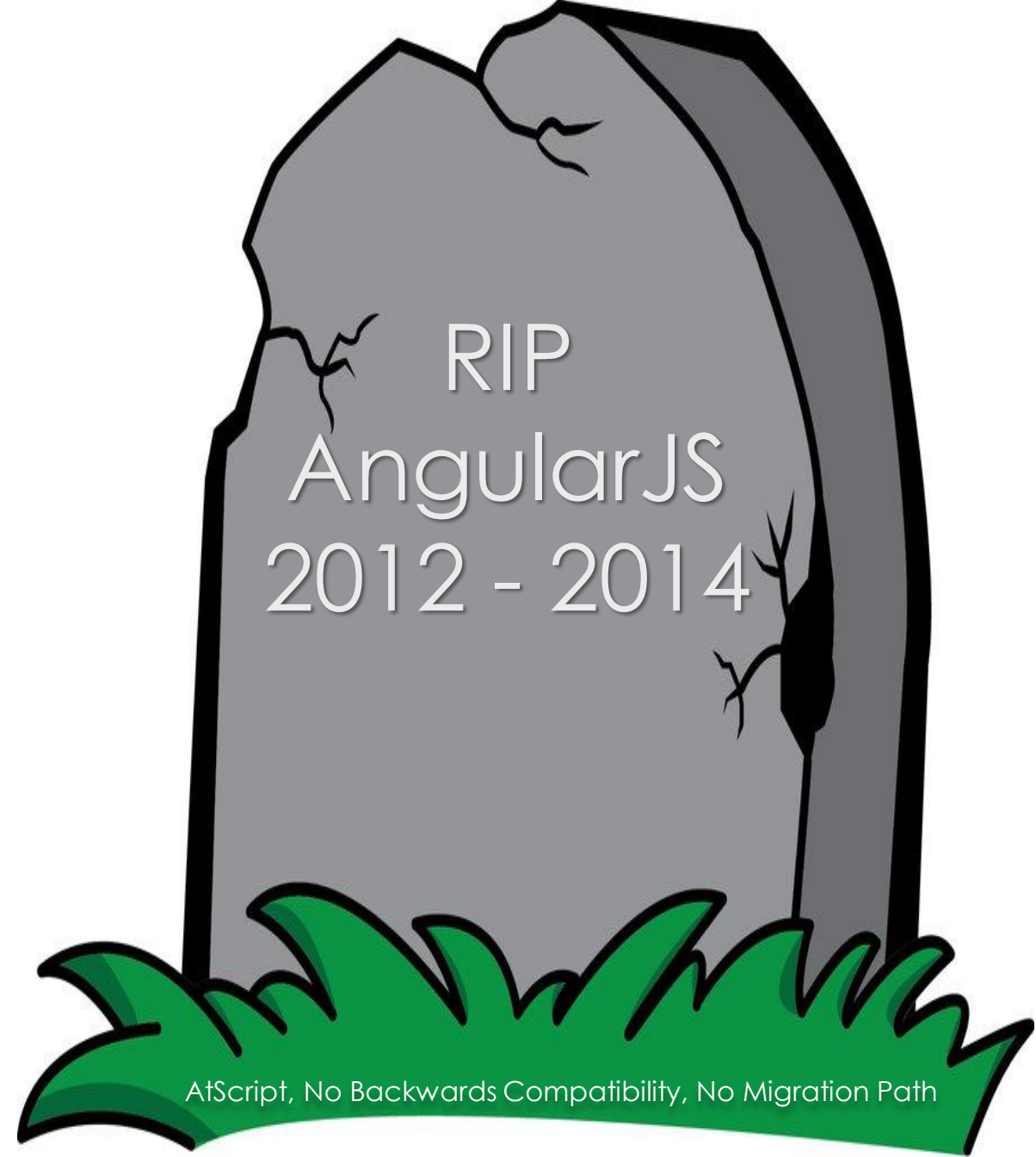
Why Aurelia instead of Angular?

”

YOUR MANAGER

Google announced
“2.0”@ ngEurope 2014

And I thought to
myself...



Somehow, this...



Angular Riots of 2014

Community in revolt...

<http://www.effectiveui.com/blog/2014/11/20/angular-2-0-a-reaction-to-the-reaction/>

<https://www.theguardian.com/info/developer-blog/2014/nov/10/on-the-ambitious-but-bumpy-road-to-angularjs-2-0>

<http://chariotsolutions.com/blog/post/angularjs-2-0-bold-new-step-typed-js-es-6-without-concerns/>

Became this...





- ▶ Angular is open source (MIT)
- ▶ Developed by Google
- ▶ Angular (2) Announced 3 years ago
- ▶ Initially used AtScript, Now TypeScript
- ▶ Initially had no upgrade path, now many
 - ▶ Including side-by-side
- ▶ Angular (2) released 8 mos ago
- ▶ Angular (4) released on March 22, 2017



- ▶ *Custom* HTML parser
- ▶ Focused on future tooling, not standards based web
- ▶ Mature CLI
- ▶ Strong community
- ▶ Lots of money
- ▶ No enterprise support from Google Core Team



aurelia

High Level Points

- ▶ Aurelia is also open source (MIT)
- ▶ Blue Spire owns Aurelia
 - ▶ Enterprise Support Available
- ▶ Founded by Rob Eisenberg
- ▶ Mature Framework (2 years)
- ▶ Current Release: Aurelia 1.1.1
- ▶ Written using modern JS (ES2015)
- ▶ Apps use either ES2015 or TypeScript



aurelia

Why good for modern apps?

- ▶ Convention over configuration
 - ▶ Focused on standards based web
 - ▶ Framework simply gets out of the way
 - ▶ Designed for developers, **not** tooling
- ▶ CLI available and evolving
- ▶ Small, but growing community



It's time to stack...

- ▶ Apples to Apples
- ▶ TypeScript to TypeScript
- ▶ Framework to Framework*
- ▶ Points of Comparison
 - ▶ Bootstrapping
 - ▶ Components
 - ▶ Elements
 - ▶ Binding
 - ▶ Services
 - ▶ Training
 - ▶ Enterprise Support

Points of Comparison

Bootstrapping

Angular ◀ ▶ Aurelia

index.html with <app>	▶ index.html with aurelia-app
main.ts for config and load	▶ main.ts for config and load
app @NgModule & class	▶ app class
1 import per route container in app.ts	▶ no imports required for pages
app.html for layout	▶ app.html for layout
app @Component & class	▶ N/A

Angular Bootstrapping

Quick Overview

```
// app.module.ts
import { NgModule } from '@angular/core';
import { BrowserModule }
    from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { RouterModule } from '@angular/router';
// Plus one import per component in this router

@NgModule({
  bootstrap: [ AppComponent ],
  imports    [ BrowserModule, RouterModule.forRoot([
    { path: 'home', component: HomeComponent }
  ])],
  declarations: [ HomeComponent, NavbarComponent ],
})
export class AppModule {
}
```

```
// app.component.ts
import { Component } from '@angular/core';
@Component({
  selector: 'app',
  templateUrl: 'app.component.html'
});
export class AppComponent {
  constructor() {}
}
```

Aurelia Bootstrapping

Quick Overview

Aurelia Class

```
// app.ts
import { Aurelia, autoinject } from 'aurelia-framework';
import { Router, RouterConfiguration } from 'aurelia-router';
import { PLATFORM } from 'aurelia-pal';

@autoinject
export class App {
  router: Router;

  configureRouter(config: RouterConfiguration, router: Router): void {
    config.options.pushState = true;
    config.map([
      { route: ['', 'home'], name: 'home',
        moduleId: PLATFORM.moduleName(""), title: 'HOME' }
    ]);
  }
}
```

- ▶ Convention over config
- ▶ Easier to read & understand
- ▶ No imports required for routed components
- ▶ HTML5 pushState not enabled by default

Points of Comparison

Components (& Pages)

Angular ◀ ▶ Aurelia

Element *@Component* + class ◀ ▶ Element class

Template Binding ◀ ▶ Template Binding

Service *@Injectable* + class ◀ ▶ Service class

Angular vs. Aurelia

Components

Angular @Component

```
// home.component.ts
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-home',
  templateUrl: './page/home.html'
});
export class HomeComponent implements OnInit {
  constructor() { }

  ngOnInit() { }
}
```

Aurelia Component

```
// home.ts

// no custom decorators needed

export class Hello {
  constructor() { }
}
```

Angular vs. Aurelia

Binding

Angular Template

```
<!-- home.component.html -->
<input [(ngModel)]="name" type="text"
        placeholder="name" />
<h3>Hobbies</h3>
<ul>
  <li *ngFor='let hobby of hobbies'>
    {{hobby}}
  </li>
</ul>
<button (click)='greet()'>OK</button>
```

Aurelia Template

```
<!-- home.html -->
<input id="name"
        type="text" value.bind="name" />
<h3>Hobbies</h3>
<ul>
  <li repeat.for='hobby of hobbies'>
    ${hobby}
  </li>
</ul>
<button click.trigger='greet()'>OK</button>
```

Angular vs. Aurelia

Services

Angular

```
import { Http } from '@angular/http';

@Injectable()
export class Data {

  constructor(private http: Http) { }

}
```

Aurelia

```
import { autoinject } from 'aurelia-framework';
import { HttpClient } from 'aurelia-fetch-client';

@autoinject
export class Data {

  constructor(private http: HttpClient) {

  }

}
```



- ▶ Installed globally
- ▶ Uses webpack under the hood
- ▶ No footprint in project
- ▶ Flexible scaffolding
- ▶ Default build pipeline



aurelia-cli

Scaffolding, Building,
Unit Testing,
Custom Tasks / Pipeline

- ▶ Installed globally
- ▶ VS Code Plugin available
- ▶ Flexible scaffolding (with Gulp) supports:
 - ▶ TypeScript, ES2015, LESS, SASS, Unit Testing
- ▶ Generator supports:
 - ▶ Components (Pages), Elements, Attributes, Binding Behaviors, Value Converters & Tasks
- ▶ Ability install / import lib to vendor bundle

Demo

▶ ng serve

▶ au run

Points of Comparison

Training

Angular ◀ ▶ Aurelia

Tons of Pluralsight and YouTube videos	◀ ▶	Some Pluralsight and YouTube videos
Tons of Books	◀ ▶	Some Books
Good documentation	◀ ▶	Good documentation
Simple examples	◀ ▶	Simple & complex examples
Google and stackoverflow confusion	◀ ▶	Google and stackoverflow works

Points of Comparison

Enterprise Support

Angular ◀ ▶ Aurelia

stackoverflow, gitter ◀ ▶ stackoverflow, gitter

Enterprise service agreements?

Not available from Core Team or Google ◀

Find a consultant: Intertech? SalsalTSoft? ◀

Enterprise service agreements?

▶ BlueSpire service agreements available
(<http://aurelia.io/support.html>, support@bluespire.com):

- ▶ Defined response times
- ▶ 1-on-1 with Aurelia Team Members
- ▶ Packaged hours

My Scorecard

Angular ◀

Big community ◀

Lots of money ◀

No core team enterprise support ◀

Heavy configuration designed for tooling ◀

Complicated, Angular specific syntax ◀

TypeScript Only ◀

▶ Aurelia

▶ Smaller community

▶ Very little money

▶ Verified core team enterprise support

▶ Simple convention with little to no config

▶ Simple, standards-focused syntax

▶ TypeScript (and JavaScript)

“

Why Aurelia instead of Angular?

”

YOUR MANAGER

What would you say now?



Ultimately, I hope that:

- ▶ You have a better sense of the differences
- ▶ I've introduced Aurelia in a positive light
- ▶ And maybe you are willing to give it a closer look



Aurelia Fundamentals

Brian Noyes, [@BrianNoyes](#) (Alexandria)

<https://www.pluralsight.com/courses/aurelia-fundamentals>

Building Applications with Aurelia

Scott Allen, [@OdeToCode](#) (Maryland)

<https://www.pluralsight.com/courses/building-applications-aurelia>

Aurelia Blog

<http://blog.aurelia.io/>

My Blog

<http://webrenaissance.org/>

Thank you.

✉ shartzog@federalidentity.com

[@SteveHartzog](#)

[#dctech](#) / [@SteveHartzog](#)

 <https://github.com/SteveHartzog>

 Aurelia Blog Skeleton (alpha; replacement for gh-pages)
<https://github.com/SteveHartzog/blog>

 VS Code Plugin – Team Essentials
<https://marketplace.visualstudio.com/items?itemName=SteveHartzog.team-essentials>